Systemantics by John Gall

It discusses the **characteristics of systems** in general(**Domain:** micro-system →very large, simple→ complex, primitive → advanced) as well as tries to build **Axiom, Axiomatic Method** to develop the science of **General Systemantics**.

Failure Mode Theorems :

1. The Fundamental Failure-Mode Theorem(F.F.T):
   Complex systems usually operate in failure mode.

2. *A complex sytem can fail in an infinite number of ways.*(derived from **Murphy's Law**)

3. *The mode of failure of a complex system cannot ordinarily be predicted from its structure.*

4. *The crucial variables are discovered by accident.*

5. *The larger the system, the greater the probability of unexpected failure.*

6. **Success** or **function** in any system may be failure in the larger or smaller systems to which the system is connected.

7. The Fail-Safe Theorem:
   When a fail-safe system fails, it fails by failing to fail safe.

- Everyting is a system.
- Everythng is a part of a larger system.
- Ther universe is infinitely systemized, both upward (larger systems) and downward (smaller systems) .
- All systems are infinitely complex. (The isllusion of simplcity comes from focussing on attention on one or a few variables.)

Efficiency :
Before one can estimate efficiency, one must first decide the **function** of the system. Since most large systems have multiple functions, many of which are not *apparent on casual inspection*, true efficiency is exceedingly difficult to estimate.

Function :
In large systems, an *intangible* attribute not susceptible to easy definition. Often equivalent to what *we think the system is doing*, or *whether we think it is doing it*

Evaluation: The process by which the system ascertains that the work it has done is genuinely good. Advanced Systems periodically review and evaluate their own evaluation procedures.

Compendium :

1. The Fundamental Theorem:
   New systems generates new problem.

2. Laws of Growth:
   Systems tend to grow, and as they grow, they encroach.

3. The Generalized Uncertainty Principle:
   Complicated systems produce unexpected out comes. The total behavior of large systems cannot be predicted.

4. *The bigger the system, the narrower and more specialized the interface with individuals.*

5. *A complex that works is invariably found to have evolved from a simple system that works.*

6. *Programs never run the first time.(Anything worth doing once will probably have to be done twice.)*

7. The Functional Indeterminacy Theorem(F.I.T):
   In complex systems, malfunction and even total nonfunction may not be detectable for long periods, if ever.

8. *Complex systems tend to produce complex responses (not solutions) to the problems.*

9. Le Chatelier's Principle:
   Complex Systems tend to oppose their own proper function.

10. The Primal Scenario:
    Systems in general work poorly or not at all.

What is a 'System'? :
System is a **set of subsystems** coordinated in some way for the achievement of some purpose.

What is a 'subsystem' then? :
This book doesn't give any clear answer on this topic, but it is implied in some way that we should apply given concepts recursively to subsystem.

Learning #1:
Nothing is more useless than strugling against a law of nature.

Learning #2:
The problem of evaluating *success* or *failure* applied to a large systems is comppounded by the difficulty of finding proper criteria for such evaluation.
In general, we can say that the larger the system becomes, the more parts interact, the more difficult it is to understand enviironmental constraints, the more obscure becomes the problem of what resources should be made available, and deepest of all, the more difficult becomes the problem of the legitimate values of the system.

Learning #3:
Goal, Purpose, Function:

- The Stated *Purpose* of the system.(The "Goal" of the designer.)
- The Built-in *Function*(What the system really does.)
- The *Goals* of the System itself.

Learning #4:
The truly pertient question is:
How does system work when its component aren't working well?

How does system fail?

How well does system function in failure mode?

Learning #5:
Systems never really do what we want them to do. So we are left out with the question What Can Be Done, given the existance of such built-in limitaions. Briefly, what can be done is to *cope*, and, on rare and satisfying occasions, to *prevail*.

Learning #6:
The greatest pitfall lies in the realm of *Problems* and *Problem-solving*. Systems can do many things, but one thing the emphatically cannot do is to Solve Problems. This is because Problem-solving is not a Systems-funtion, and there is no satisfactory Systems-approximation to the solution of a Problem. A System represents someone's solution to the problem. The system does not *solve* the problem.

Learning #7:
The undertaking should be begun only where the present evil is very clear and the consequences of a miscarriage are juged to be no more unbearable than a continuation of the original unsatisfactory situation.